Alien Invasion

Oh no! Alien pixel spaceships are descending on the Minecraft world! You'll have to pilot a pixel spaceship of your own and fire pixel bullets to stop them!
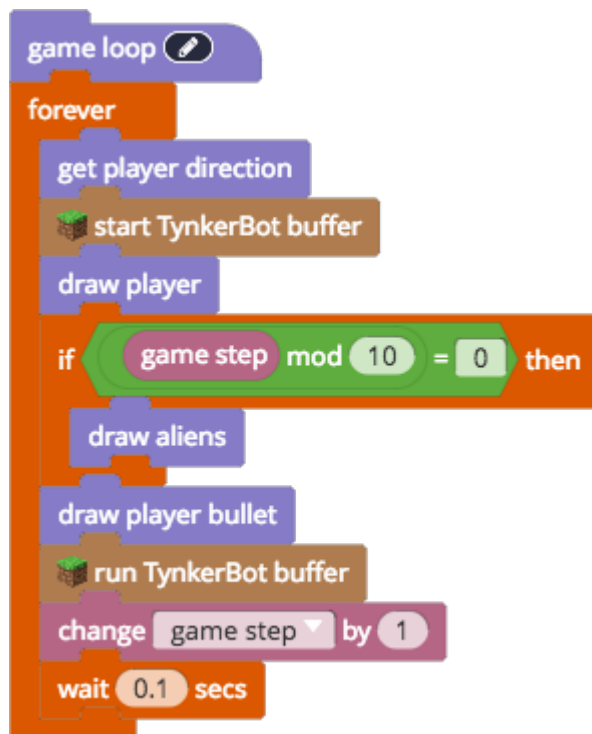
In this project, you will recreate a classic arcade game. You'll use Minecraft blocks to simulate the iconic pixelated appearance of older video games.

Use the bot to place and remove the elements of the game, detect the presence of a "button" to steer the player's ship, and manage everything in a game loop that runs periodically.

How the Game Works

To recreate this classic game in Minecraft, you will "draw" the visuals of the game--ships and bullets--with blocks. You need to repeatedly draw and redraw these blocks as the game's state changes. That way you can animate the movement of the player's ship, enemy ships, and bullets.

In the code given to you, look for the "**game loop**" function:

```
game loop ✎
forever
    get player direction
    🟫 start TynkerBot buffer
    draw player
    if ( game step mod 10 = 0 ) then
        draw aliens
    draw player bullet
    🟫 run TynkerBot buffer
    change game step ▾ by 1
    wait 0.1 secs
```

Once this function is called, it repeatedly does several things over and over, several times per second:

- Determines which direction the player's ship should move next
- Starts accepting instructions for the Minecraft bot, which is used for all "drawn" elements
- Draws the player's ship
- Draws the aliens' ships, but only every 10th time through the loop
- Draws the bullet that the player's ship has fired
- Sends all bot/drawing instructions to the server
- Adds 1 to a counter that keeps track of how many times the loop has run
- Waits for a short period

Try playing this game as is. Punch the glass blocks to move the player ship (the emerald box) left and right. The player ship will fire at the alien ships (redstone boxes) automatically.
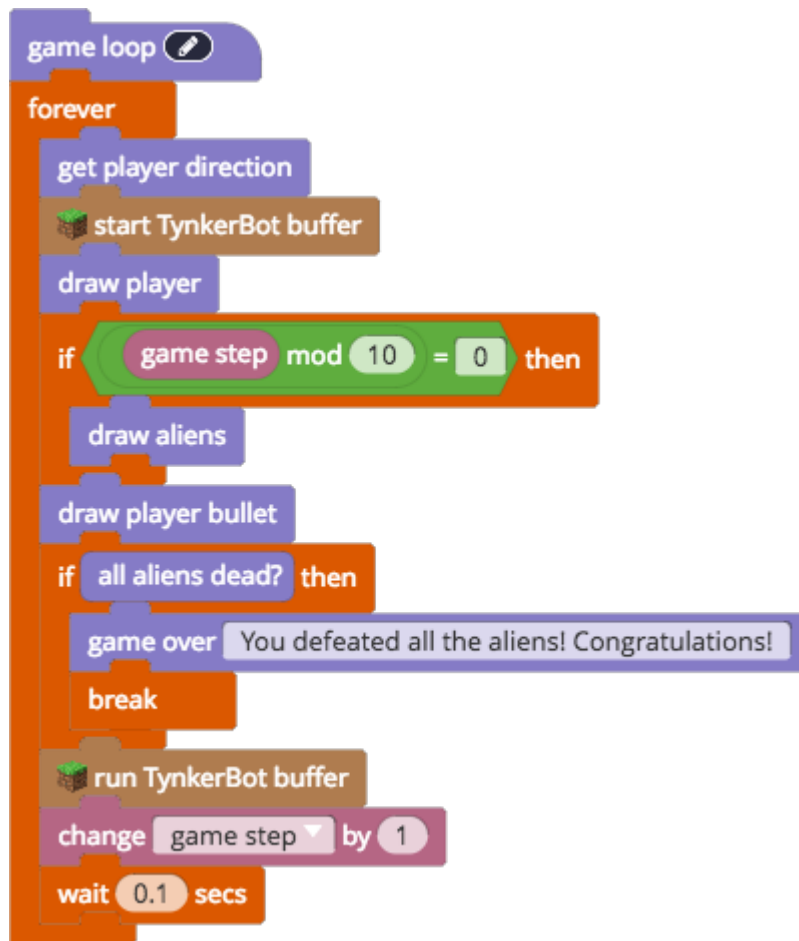
Add the Win Condition

As it is right now, the "**game loop**" function will keep running forever. There's no point to leaving the game running after the player has beat it, though!

Let's add a "win condition" to the loop so that if the player has won, the game ends.

1. Drag the above "if-else" block into the loop, in between "draw player bullet" and "change by."

2. Inside the "if-else" block, attach a "game over" function block. For the game over message, write something like "Congratulations, you won!"

3. After the "game over" block, attach a "break" block. This will stop the game loop.

Your edited code might look like this:

```
game loop ✏

forever
    get player direction
    🟫 start TynkerBot buffer
    draw player
    if ( ( game step  mod  10 )  =  0 )  then
        draw aliens

    draw player bullet
    if  all aliens dead?  then
        game over  You defeated all the aliens! Congratulations!
        break

    🟫 run TynkerBot buffer
    change  game step ▾  by  1
    wait  0.1  secs
```

Once you've added this code to your game loop, try playing the game again. After you beat the game, you should see the game area vanish and a victory message appear.

Add a Challenge

So far, you can shoot down the alien ships, but there is no danger to you! Let's make the game more dangerous by making the alien ships fire back.

```
draw alien bullet
```

Drag the "**draw alien bullet**" function into the game loop, right after the "**draw player bullet**" function, so that the loop looks something like this:

Test your game. Notice that the first alien ship will periodically fire a bullet downwards toward the player.

Add a Loss Condition

The aliens are attacking now, but there's no danger yet; the alien bullet doesn't actually harm the player ship. Let's fix this so the game is actually challenging.

1. Drag the above "if-else" statement into your game loop, after the "draw alien bullet" function. Its conditional detects whether the player has not been flagged as "alive" (i.e. the player has been hit/"killed") in the game's background code.

2. Inside the "if-else" block, add another "game over" block.

3. This time, add a loss message instead of a win message.

4. Add a "break" block next so that the loop will stop running.

Your code should look something like this now:
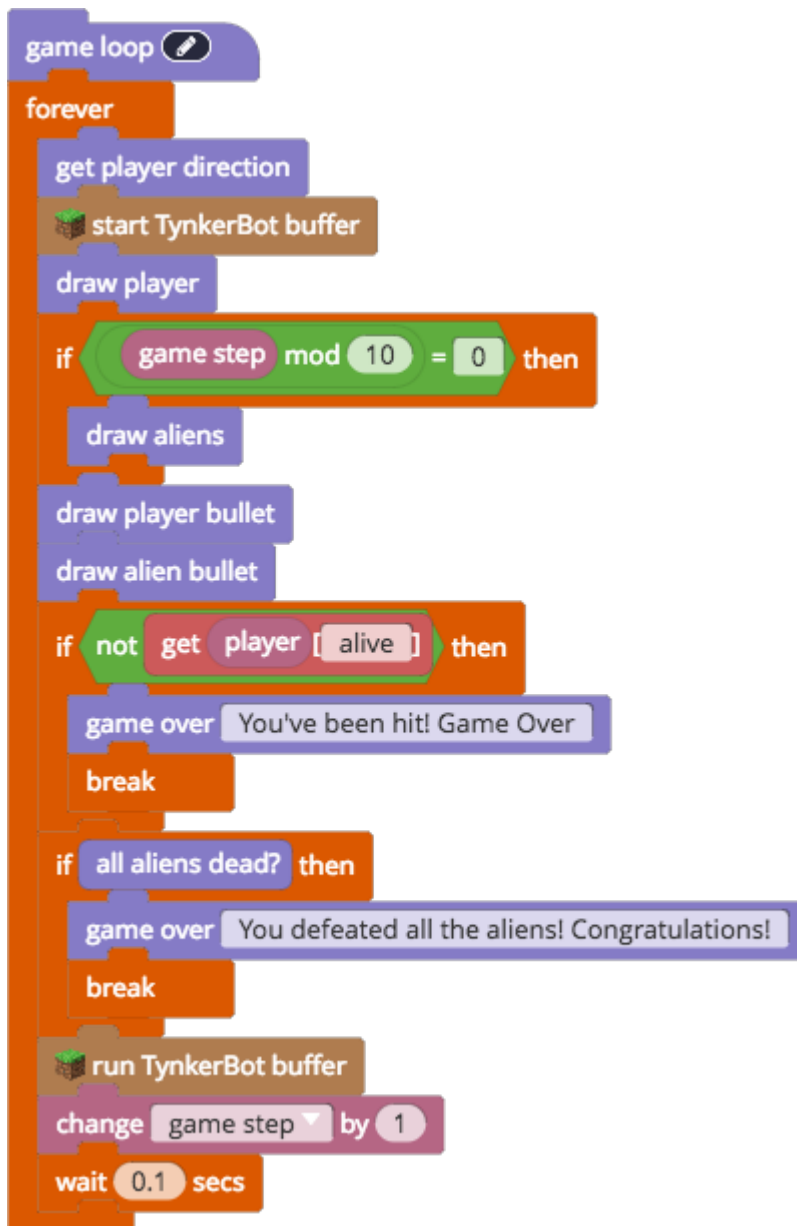
Test your game. Now, if you allow the player ship to be hit by an alien bullet, the game should disappear and display a loss message.

Change the Aliens' AI

You might have noticed how it is always the first alien that fires a bullet at the player, even if that alien has already been destroyed!

Let's make this behavior more interesting, so that it is always a random (living) alien that fires.

Scroll down to the **"draw alien bullet"** function at the very bottom of the code area. It's complex, but don't worry too much about how it works. Find the last section of the function, which looks like this:
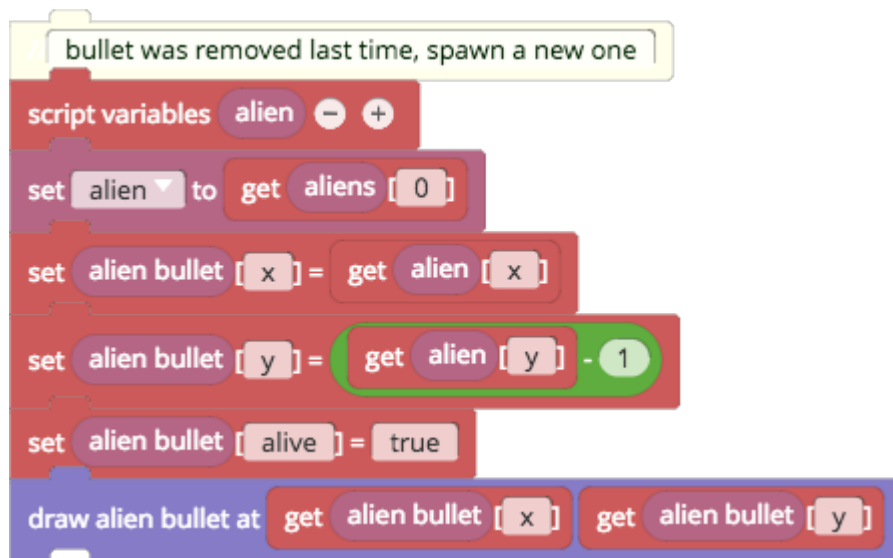
1. Find the code block in this section that sets the "alien" variable to "get aliens[0]."

2. Remove the "get aliens[0]" and discard it.

3. Replace it with the "get random alien" function that we have provided you:

The code section should now look like this instead:
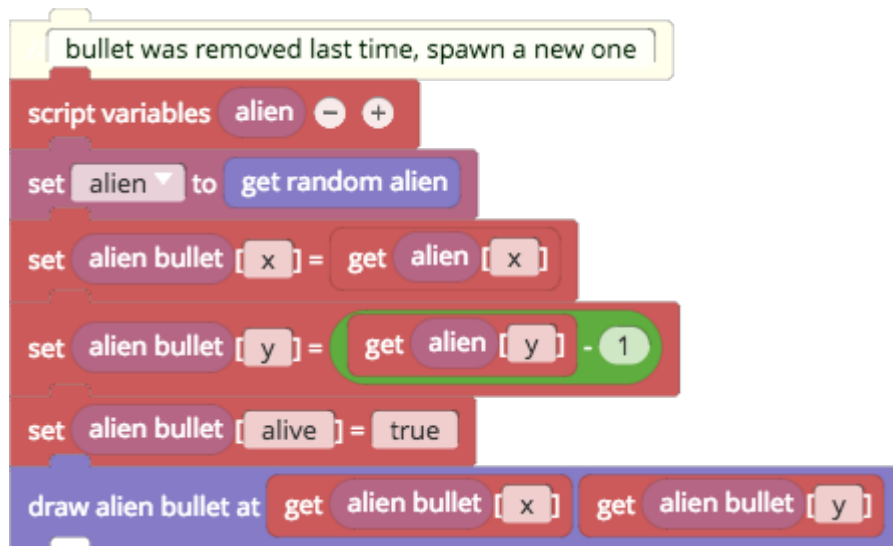
Test your game. Notice that different aliens fire at the player now!

Bonus: Make the Game Your Own

Awesome! Your game is working pretty well now. Have fun playing it!

If you feel brave, you can play with the game's underlying code and try to make even more improvements, such as...

- Adding a second "game over" condition: when the aliens reach the ground, you lose. Try checking the value of the "game step" variable.
- Enhancing the "draw alien ship" and "draw player ship" functions to make more interesting ships. Make sure to adjust "erase entity" as well.
- Increasing the number of aliens created in the "game setup" function.
- Challenge: Adding more alien bullets.
- Challenge: Adding "shields" above the player ship.